

CONSTRAINT REASONING FOR REGION-BASED IMAGE LABELLING

C. Saathoff

ISWeb - Information Systems and Semantic Web
University of Koblenz
D-56070 Koblenz, Germany
saathoff@uni-koblenz.de
Fax: +49(261)2872721

Keywords: Semantic Multimedia, Image Understanding, Constraint Reasoning

Abstract

In this paper we present an approach to exploit explicit spatial knowledge for the labelling of image regions. We employ a two-step process, consisting of an initial labelling carried out by a segment classification algorithm and a second step refining the initial labelling based on knowledge about the spatial arrangements of concepts. For the second step we transform the initially labelled image into a *Constraint Satisfaction Problem* and employ standard *Constraint Reasoning* techniques to ensure the spatial consistency of the labels. We provide an evaluation and some hints on future research directions.

1 Introduction

Digital media is omnipresent in our every-day work with computers, and automatic methods to help in annotating, organising and retrieving the content seem desirable. Research on image understanding and multimedia analysis in general has a long history, and good results are achieved for numerous problems, such as person detection or scene classification. It has however been noted several times that methods based on low-level features do not suffice to provide for more generic annotations and that the integration of explicit background knowledge is required [9].

In this paper we present an approach to exploit explicit background knowledge for the labelling of image regions. The main focus of this approach is to find a consistent labelling of an image that is useful for later retrieval, and to provide algorithms that are domain independent. The domain specific knowledge is explicitly represented in a domain ontology. In a first implementation we specifically focused on the usage of spatial knowledge applied on top of a segment labelling algorithm.

The remainder of this paper is organised as follows. After shortly presenting related work in the next section, we will

outline the environment, in which the presented work was developed, in Section 3. In Section 4 we will give an overview of the relevant background of constraint satisfaction problems and constraint reasoning that is needed to follow the description of our work in Section 5. Our first implementation and evaluation results are discussed in Section 6. We finally conclude with an outlook on future work in Section 7.

2 Related Work

A number of approaches have been represented that employed explicit spatial knowledge for automatic annotation of images. In [5] a system is presented that allows domain experts to define rules on primitive visual features, such as colour or simple shapes, in order to annotate images automatically. The system was employed for automatic annotation of fuel cells. Their work strongly requires a direct mapping of low-level features to semantic features, which is feasible within scientific domains, such as microscopy images, but not in the domains we consider.

In [3] a configuration system is adopted to provide high-level scene interpretations. The system is evaluated on so-called table laying scenes. The domain models are based on the spatial arrangements of the concepts found within this domain. In [4] a complete platform for cognitive vision is presented. It consists of one knowledge base for supervising the low-level detectors, which detect specific features in the content, one knowledge base for anchoring concepts in the content and one knowledge base for interpreting the image based on the concepts found. These approaches represent a class of systems that try to annotate multimedia content fully driven by concise domain models often defined in ontologies. These system already evolved in the nineties [6, 8] and often rely on complete domain models. A goal of our approach is that less complete domain models are sufficient, so that the application within more generic domains becomes feasible.

Finally, an interesting approach is presented in [10]. There, images are annotated semi-automatically. A user can manually prune the search space by specifying hints such as "A L-shaped building in the upper left corner." A constraint reasoner is employed to take care of enforcing the user hints. This approach is most similar to our work but obviously does not build upon pre-segmented images and is rather focused on

object detection.

3 Knowledge-Based Image Analysis

The work presented in this paper was developed within the aceMedia project¹. The goal of aceMedia is to allow for novel access to multimedia introducing the *Autonomous Content Entity (ACE)*. One of the core functionalities is the self-annotation capability of an ACE, i.e. the potential to detect the content of the data and to provide a semantic annotation.

In order to achieve this goal, a number of multimedia analysis algorithms are available for images. Namely the project provides scene classification, person and face detection, face recognition, natural language processing in order to analyse textual annotations and a knowledge-based segment classification. The goal of the multimedia reasoning within aceMedia is the integration of the provided analysis results into a consistent high-level annotation of the image. Several modules are developed within the project in order to exploit the manual annotations as contextual knowledge, to derive higher level annotations from the object detection and segment classification results, and finally checking for the semantic consistency of the annotation. The latter module is described in this paper.

The result of the analysis chain is a *region-based annotation* of the image, i.e. an annotation of regions of the image that refer to concepts detected by one of the analysis algorithms. Within this paper we only employ the knowledge-based segment classification and apply our approach in order to check for and ensure the spatial consistency of the provided labels. The overall analysis chain consists of the following steps:

1. segmentation of the input image
2. extraction of spatial relations between segments
3. assigning a set of possibly depicted concepts with a degree of confidence to each segment, the so-called hypothesis sets
4. applying spatial constraints in order to remove inconsistent concepts from the hypotheses sets

The first steps can be handled by existing techniques, namely image segmentation, extraction of spatial relations, and classification. The third step is usually the most problematic in image understanding, since it tries to derive the semantics from the low-level features. However, we do not require this step to produce absolutely accurate results, since we are just interested in a list of labels that are possibly being depicted and a degree of confidence for each label. We call this set the *hypothesis set*

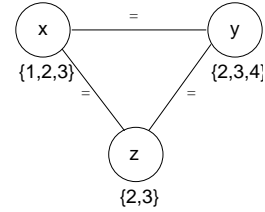


Figure 1: A simple constraint satisfaction problem.

of the segment. This specifically allows us to use light-weight and less complex algorithms for the classification and to improve the results afterwards. The infrastructure, including the ontology infrastructure and a more thorough description of the knowledge-based segment classification can be found in [7].

In this paper we specifically try to solve the last step, i.e. identification of the labels being inconsistent with regard to labels of spatially related segments. Although we only use the segment classification algorithm so far, our main target is to provide an approach that combines the outputs of several object detection and segment classification algorithms into a single consistent annotation.

4 Constraint Reasoning

In the following we will introduce *Constraint Satisfaction Problems (CSPs)* first informally based on a small example and then formally based on [1]. Informally, a constraint satisfaction problem consists of a number of variables and a number of constraints. A variable is defined by its domain, i.e. the set of values that can be assigned to the variable. A constraint relates several variables and thereby restricts the legal assignments of values to each of the involved variables. *Constraint Reasoning* is the process of computing a solution to the given CSP, i.e. an assignment of values to the variables that satisfy all the given constraints on the variable.

In Fig. 1 a simple CSP with three variables x , y , and z and three constraints are depicted. The domains of x , y and z are $D(x) = \{1, 2, 3\}$, $D(y) = \{2, 3, 4\}$ and $D(z) = \{2, 3\}$. The constraints are $x = y$, $x = z$ and $y = z$, so that in a solution to the problem, the values of x , y and z must be equal.

During constraint reasoning, usually two processes take place. First *local consistency* is checked and then a *search* is carried out. Local consistency only considers subparts of the CSP and removes such values from the domains of the considered variables, that do never satisfy the constraints. Search then evaluates possible assignments and checks whether a solution can be found.

In the example, the domain of x would be reduced to $D(x) = \{2, 3\}$ and the domain of y to $D(y) = \{2, 3\}$ based on the

¹<http://www.acemedia.org/>

constraints. E.g. an assignment of 1 to x would in every case violate the constraint $x = y$, since 1 is not a member of $D(y)$. However, this step does not mean that a solution to the CSP is found solely by assigning an arbitrary value from the domain to the variable. Assigning $x = 2, y = 3, z = 3$ would apparently not be a solution, although all values are members of the domain. A valid solution like $x = 2, y = 2$ and $z = 2$ is then found using search techniques such as backtracking.

A formal definition of a CSP, based on [1], consists of a set of variables $V = \{v_1, \dots, v_k\}$ and a set of constraints $C = \{c_1, \dots, c_l\}$. Each variable v_i has an associated domain $D(v_i) = \{i_1, \dots, i_m\}$, which contains all values that can be assigned to v_i . Each constraint c_j relates a number of variables $v_1, \dots, v_r \in V$. A constraint c_j is defined to be a relation on the domains of the related variables, i.e. $c_j \subseteq D(v_1) \times \dots \times D(v_r)$. The constraint is said to be solved, iff both $c_j = D(v_1) \times \dots \times D(v_r)$ and c_j is non-empty. A CSP is solved if both all of its constraints are solved and no domain is empty, and failed if it contains either an empty domain or an empty constraint.

We will not further elaborate on local consistency notions and search techniques, since they are out of the scope of the paper. Standard methods exist to solve a given CSP, and currently we employ such standard techniques. However, best performance is usually achieved by employing domain specific methods, often augmented by domain heuristics. We refer the reader to the extensive literature on the subject [1, 2] in order to get an idea of existing methods. Once the labelling performance is in a stable and satisfactory state, we will work on local consistency and search techniques appropriate for the specific application of CSPs to the image labelling problem.

5 Image Labelling as a Constraint Satisfaction Problem

In this section we describe how to transform an initially labelled image into a constraint satisfaction problem that can subsequently be solved with standard constraint reasoning techniques. The core idea of our approach is to identify and remove those labels from the hypothesis sets of the image that are inconsistent with respect to the spatial arrangements usually found in images within a given domain. Thus, during constraint reasoning we would like to exploit knowledge about the typical spatial arrangements found in images within a certain domain. For instance, within the beach domain, such spatial knowledge could model that sky is usually depicted above all other concepts, or that one will not find any sea left of the sky. Also stronger heuristics might be useful. A person is e.g. usually not fully surrounded by sky on an image. In principle, such knowledge, even if heuristic in nature, can help to disambiguate labellings, but will also constrain the number of possible interpretations.

Obviously, the kind of rules explained above can naturally be

represented as constraints. Consider for example two regions A and B related by a spatial relation *left-of*, such that A *left-of* B . Let the label for A be *Sky* and let B be labelled with *Sky* and *Sea*. We can now view the above situation as CSP treating the segments as variables, the labellings as their domains and the relation as a constraint on the variables. A constraint would need to constrain the number of valid assignments to both variables, effectively removing *Sea* from the labelling for B .

Treatment of the image labelling problem as a constraint reasoning task apparently comprises two core parts: the representation of spatial knowledge as constraints and the transformation of a segmented and initially labelled image into a CSP. In the following we will first discuss how to represent spatial knowledge as constraint definitions and then present the algorithm that generates a CSP out of the initially labelled image. The resulting CSP be then be solved using standard constraint reasoning algorithms. We conclude this section with a small example explaining the ideas introduced.

5.1 Spatial Background-Knowledge as Constraints

The spatial background knowledge we use is supposed to encode the valid spatial arrangements of concepts within a certain domain. Currently we only support absolute and relative binary spatial constraints. These directly translate to unary respectively binary constraints. An absolute spatial relation expresses the absolute spatial position of a segment in an image, such as *above-all* and *below-all*, expressing that a segment is on the very top or the very bottom of an image. Relative spatial relations represent the position of a segment with respect to another segment. Examples are *left-of* or *above*.

A constraint c on a set of variables $\{v_1, \dots, v_n\}$ is defined as a relation on the variable domains, i.e. $c \subseteq D(v_1) \times \dots \times D(v_n)$. With every constraint we associate a type $type$, which refers to specific spatial relation represented. We will denote a spatial relation of a given type from now on as c_{type} . With each constraint type we associate a set of tuples of domain concepts, which we call the domain of the constraint type, denoted $D(c_{type})$. Let C be the set of all domain concepts, we then define the domain as $D(c_{type}) := \{(l_1, \dots, l_n) | (l_1, \dots, l_n) \in C^n\}$, with l_1, \dots, l_n being concepts forming a valid arrangement with respect to a spatial relation of the type $type$. A constraint c_{type} on a set of variables $\{v_1, \dots, v_n\}$ can now be defined as $c := \{(l_1, \dots, l_n) | (l_1, \dots, l_n) \in D(c_{type})\}$.

5.2 Transforming a Labelled Image into a CSP

Now, to describe the transformation, we will shortly introduce a formal model for a labelled image. A labelled image is a tuple $I = (S, SR)$, where S is the set of segments produced by

the initial segmentation and SR is the set of spatial constraints extracted by the spatial extraction module. For each segment $s \in S$ the set of labels is denoted as $ls(s)$. The set of all possible labels is named C and $ls(c) \in C$ must hold. Each spatial relationship $r \in SR$ is a tuple $(type, (s_1, \dots, s_n))$, where $type$ determines which spatial relationship the tuple represents and (s_1, \dots, s_n) is the set of segments the spatial relationship is defined on.

The segmented image and the spatial relations between the different segments are directly transformed into a CSP by instantiating a variable for each segment and adding a corresponding constraint for each spatial relation between two segments. Obviously, the hypotheses sets become the domains of the variables, so that the resulting CSP is a finite domain CSP. Two types of spatial constraints can be distinguished: relative and absolute. Relative spatial constraints are binary constraints and derived from spatial relations that describe the relative position of one segment with respect to another one, like *left-of* or *above-of*. Absolute spatial constraints are derived from the absolute positions of segments on the image, like *above-all*, which describes that a segment is on the top of the image. These are apparently unary constraints.

Let $I = (S, SR)$ be an labelled image as introduced above, then the transformation itself can be carried according to the subsequent algorithm:

1. For each segment $s \in S$ create a variable v^s .
2. Set $D(v^s) = ls(s)$.
3. Let SC be the set of all spatial constraints defined in the domain knowledge, then
 - (a) add for each absolute spatial relation $r \in SR$ on a segment s with $r = (type_r, \{s\})$ a unary spatial constraint c_{type_r} on the variable v^s of the according type.
 - (b) add for each relative spatial relation $r \in SR$ between two segments $s, t \in S$ with $r = (type_r, \{s, t\})$ a binary spatial constraint $c_{type_r} \in SC$ between the variables v^s and v^t of the according type.

The result is a CSP conforming to what was introduced in Section 4. Standard constraint reasoning techniques can now be used to solve the CSP, and because of the finiteness of the problem, all solutions can be computed. The latter property is not exploited currently, but we intend to employ fuzziness and provide an ordered list of possible interpretations as an output in a future version. Further, it should be noted that we did not research into heuristics and optimal constraint reasoning approaches for our problem. Therefore, we are using standard algorithms currently.



Figure 2: The input image.

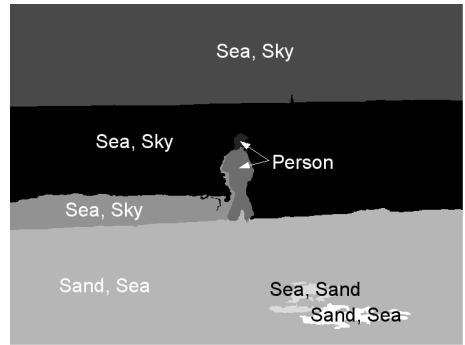


Figure 3: Segmented and initially labelled image.

5.3 Example

An example is depicted in Figures 2 to 5. The input image is depicted in Figure 2 and the initial labelling in Figure 3. The annotation produced by the segment labelling algorithm is shown in Figure 4. In this case the labels with the highest score were kept. It is easy to see that two errors were made by the segment classification. The topmost segment was labelled as *Sea* instead of *Sky* and one of the small segments within the sand regions was labelled with *Sea*.

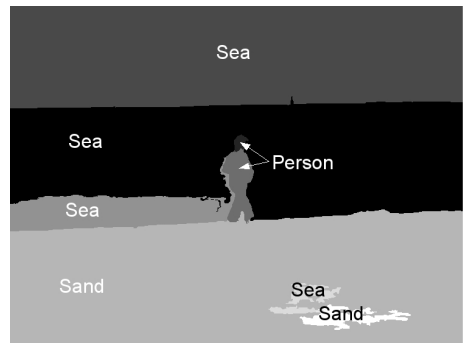


Figure 4: Image with highest-scoring labels.

After applying the constraint reasoning we obtain the image labelling depicted in Figure 5 and the erroneous labellings were both corrected. For the topmost segment the absolute spatial relation *above-all* restricts the segment to the label *Sky* and

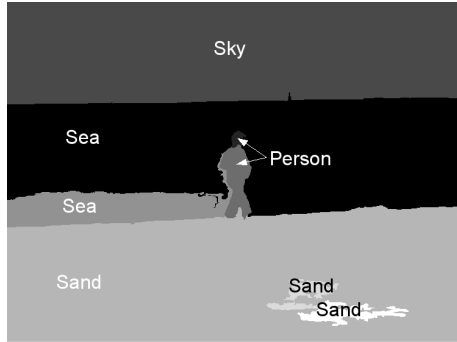


Figure 5: Labelling after application of the constraint reasoner.

the second wrong label was corrected using the *contained-in* constraint that does not allow a *Sand* segment to contain a *Sea* segment.

6 Evaluation

We implemented the proposed approach using the *Java Constraint Library (JCL)*². The initial and the resulting labellings are represented using the ontology infrastructure presented in [7]. We then carried out an evaluation. In this section we will give a short overview of our evaluation method, the setup, and then discuss the evaluation results. Since our method currently aims at improving the segment classification, we first evaluated the segment classification algorithm on the test data and then the classification and the constraint reasoning combined. Thus, we were able to assess the performance improvement reached by using the explicit spatial knowledge. In both cases we used the latest version of the algorithm presented in [7].

The ground truth was defined on a 8x8 grid that was layered on the image and each grid cell was labelled with the concepts contained in that cell. In this evaluation we used the concepts *Sky*, *Sand*, *Sea* and *Person*. We then mapped each segment produced during segmentation to the grid cells overlapping the segments. If a grid cell contained a label x and a segment mapped to that grid cell also was labelled with x , we counted that as a correctly found label (cf). Because our evaluation is based on the grid, we counted for each cell whether a label was found or not. This means, if a segment was overlapped by two cells, we counted the labels of the segment twice. Finally we counted the number of times a label was present in the ground truth, i.e. again for each cell and not for solely for the segments. We then defined three well know metrics from the information retrieval field: precision, recall and the F-Measure. Precision was defined to be $p = \frac{cf}{f}$, recall was defined as $r = \frac{cf}{c}$, and the F-Measure was computed the standard way, i.e. $F = \frac{2pr}{p+r}$.

We created ground truth for over 50 images from the beach

²<http://liawww.epfl.ch/JCL/>

Concept	Precision	Recall	F-Measure
Sky	0.77	0.69	0.73
Sea	0.66	0.59	0.62
Sand	0.75	0.94	0.84
Person	0.33	0.65	0.41
Total	0.69	0.75	0.72

Table 1: Evaluation results for the segment classification.

Concept	Precision	Recall	F-Measure
Sky	0.78	0.91	0.84
Sea	0.73	0.53	0.62
Sand	0.85	0.97	0.9
Person	0.38	0.62	0.47
Total	0.76	0.82	0.78

Table 2: Evaluation results for the constraint reasoner.

domain. The results in Table 1 for the segment classification already show good performance. However, in Table 2 the results of the constraint reasoning approach is shown, and for most concepts an improved performance was achieved. The largest improvement was achieved for the concepts *Sky* and *Sand*, which is due to the global constraints *above-all* and *below-all* restricting the top-most and bottom-most segment to *Sky* and *Sand*, respectively. These heuristics provide good results and give a hint that heuristic knowledge seems to be valuable for image interpretation. The other constraints, however, were only of use in a smaller number of cases. The segmentation was limited to eight segments for each image, which proved to be rather low. In principle an over-segmentation would be desirable in order to produce also segments for smaller regions, but the over-segmentation provided by the segmentation used in our implementation however did not provide good over-segmentation results. Given an over-segmentation, we expect the spatial constraints to have a higher impact.

The overall performance was improved by approximately 10%, which seems promising with respect to the known limitations of the current implementation. Further, it shows that the introduction of spatial knowledge into the labelling process leads to better performance. In the following section we will outline our future plans to provide a more robust implementation.

7 Future Work and Conclusions

In this paper we presented an approach to exploit explicit knowledge about spatial arrangement of concepts in images in order to improve a region-based labelling. We employ an initial labelling of the image provided by a segment classification

algorithm, providing a list of hypotheses for each region. We then remove those labels from the image that are inconsistent with respect to their spatial arrangement. The final annotation is created by keeping the highest scored label for each image. Through an evaluation we showed that the approach is feasible in order to refine and improve the labelling of a pure pattern based labelling algorithm. However, the evaluation also showed some weaknesses of our approach, which we will discuss in the following. Because the results showed to be promising in general, we plan to continue research on this approach and provide a more robust and scalable version in the future.

The evaluation showed that mainly the absolute spatial constraints lead to improved the results. These constraints reduce the search space significantly, but in turn they are also very heuristic and often do not apply. In general hard heuristics seem to be ideal to produce good interpretations, but in order to cover a large part of the feasible interpretations, an intelligent way of applying the heuristics has to be found. We will research into soft constraints in order to find a more flexible way of applying constraints. This will allow us to state preference among constraints, and to apply constraints incrementally. The introduction of fuzziness will additionally allow us to produce an ordering among the possible interpretations. This way, we hope to select the best interpretations based on some kind of global evaluation measure. In general we hope that this leads to the representation of different *domain models*, stating commonly found spatial arrangements of concepts in images. The flexible application of the different domain models for the image understanding task, and the global evaluation measure will ideally lead to an algorithm that can find the best fitting domain model for a given image, providing an optimal annotation.

Another important aspect of our research is the integration of image analysis results on a metadata level. We therefore will integrate the output of a person detection and face detection algorithm in to our approach. Since these highly specialised algorithms provide good results for the detection of persons respectively faces, we expect to have higher overall precision and recall for those concepts, in the end improving the overall performance of the system and resulting in a better generic labelling of the image.

Finally, the acquisition of knowledge is still not solved. Currently the spatial knowledge is defined explicitly. But in general specifying spatial constraint by hand on large domain ontologies is a very tedious task. Especially when preference among constraints has to be stated and meaningful domain models have to be found, a manual definition seem not to be feasible. We will therefore look into ways of learning the according knowledge from examples, possibly integrated in an interactive environment.

Acknowledgements This research was partially supported by the

European Commission under contract FP6-001765 aceMedia. The expressed content is the view of the authors but not necessarily the view of the aceMedia project as a whole.

References

- [1] Krzysztof R. Apt. *Principles of Constraint Programming*. Cambridge University Press, 2003.
- [2] Roman Barták. Constraint programming: In pursuit of the holy grail. In *Proceedings of Week of Doctoral Students (WDS99)*, pages 555–564, 1999.
- [3] Lothar Hotz and Bernd Neumann. Scene interpretation as a configuration task. *Künstliche Intelligenz*, 3:59–65, 2005.
- [4] Céline Hudelot and Monique Thonnat. A cognitive vision platform for automatic recognition of natural complex objects. In *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2003)*, 2003.
- [5] Suzanne Little and Jane Hunter. Rules-by-example - a novel approach to semantic indexing and querying of images. In *International Semantic Web Conference*, pages 534–548, 2004.
- [6] Heinrich Niemann, Gerhard F. Sagerer, Stefan Schröder, and Franz Kummert. Ernest: a semantic network system for pattern understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):883–905, 1990.
- [7] Kosmas Petridis, Stephan Bloehdorn, Carsten Saathoff, Nikos Simou, Vassilis Tzouvaras, Siegfried Handschuh, Yannis Avrithis, Yiannis Kompatsiaris, Steffen Staab, and Michael G. Strintzis. Knowledge representation and semantic annotation of multimedia content. *IEE Proceedings on Vision Image and Signal Processing, Special issue on Knowledge-Based Digital Media Processing*, 153(3):255–262, 2006. to appear.
- [8] T. Russ, R. MacGregor, B. Salemi, K. Price, and R. Nevatia. Veil: Combining semantic knowledge with image understanding. In *ARPA Image Understanding Workshop*, 1996.
- [9] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, 2000.
- [10] Rohini K. Srihari and Zhongfei Zhang. Show&tell: A semi-automated image annotation system. In *IEEE MultiMedia*, volume 7, pages 63–71, 2000.



Click here to return to
Session

Supported By:

The Microsoft logo, featuring the word "Microsoft" in a white, sans-serif font on a blue rectangular background.

The Motorola logo, consisting of a stylized 'M' inside a circle followed by the word "MOTOROLA" in a bold, sans-serif font.

Media Partners:

The logo for Deccan Herald, featuring the word "DECCAN" in a blue, serif font, followed by a small icon of a person and the word "HERALD" in a blue, serif font.

The logo for CIOL (Computer Information Online), featuring the letters "CIOL" in a large, bold, sans-serif font, with a globe icon integrated into the letter 'O'. Below the logo is the tagline "MAKING IT YOUR ADVANTAGE" in a smaller, sans-serif font.